WEB DEVELOPMENT USING MERN STACK

HTML

HTML BASICS

Html is a skeleton of a web page

Html is not case sensitive

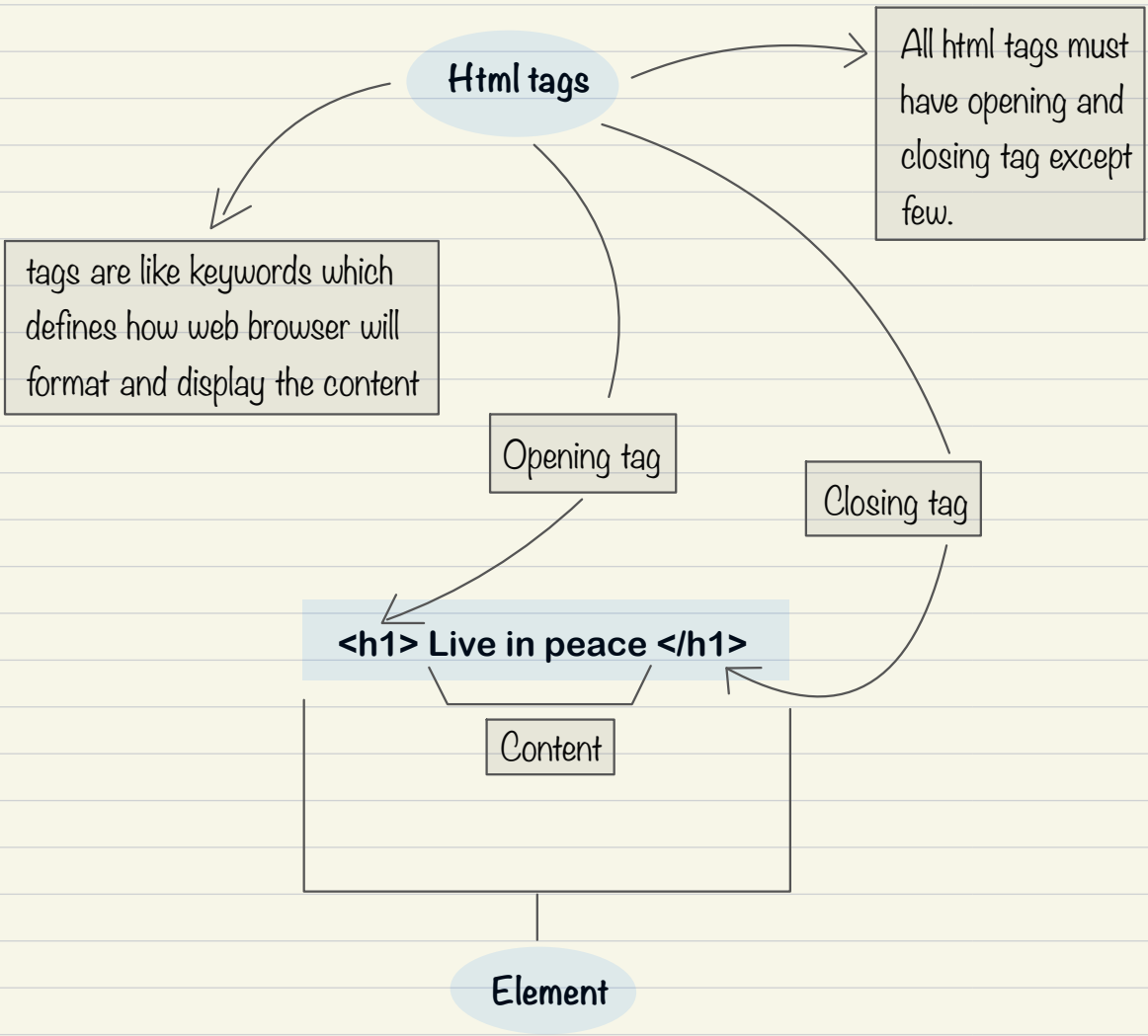
HTML

Hyper Text Markup Language

HyperText refers to links that connect web pages to one another

It uses "markups" to annotate content for display in browser

Html is not a programming language. It is a markup language



Note: Some HTML elements have no content (like the <br> element). These elements are called empty elements. Empty elements do not have an end tag!

Note : to comment in html use this : <!-- Content -->

Html text element

Block elements

A block level element always starts on a new line and the browser automatically add some space before and after the element

A block level element always takes up full width available i.e. stretches out to the left and right as far as it can

Example : , <p> , <h1>

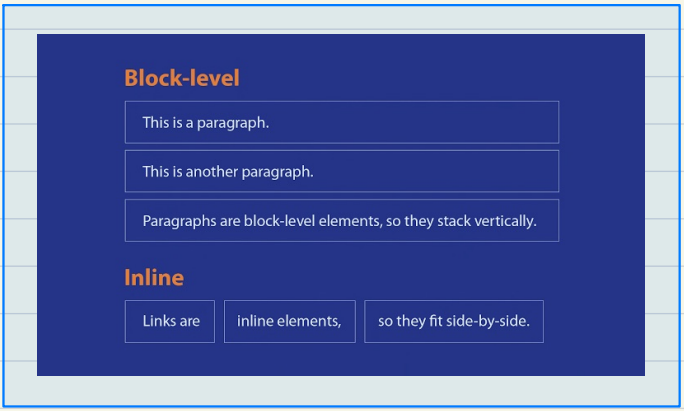
Inline elements

An inline element does not start on a new line

An inline element only takes up as much width as necessary

Example : , <a>

Note : An inline element cannot contain a block level element



Header tag :

Block level element

<h1> heading 1 </h1>
<h2> heading 2 </h2>
<h3> heading 3 </h3>
<h4> heading 4 </h4>
<h5> heading 5 </h5>
<h6> heading 6 </h6>

We can have total h1 to h6 header elements, h1 being the biggest and h6 being the smallest

<header> cannot be placed within another <header> element

Note: Browsers automatically add some white space (a margin) before and after a heading.

Paragraph tag :

Block level element

```
<p>This is a Paragraph</p>
```

---

 tag :

Block level element

```
<div class="warning" id="red-alert">
<p> Name is <strong> Chauhan Satyam </strong>
</p>
</div>
```

Generic container, no effect on the content until styled using css

The div element should be used only when no other semantic element is appropriate

---

tag :

Inline level element

```
<p> I am a proud student of
<span class="C1">Dot Batch</span>,
<span class="C2">Supreme Batch</
span> taught by <em>Love Babbar</em>.
</p>
```

generic container for phrasing content, which does not inherently represent anything

span element should be used only when no other semantic element is appropriate

---

<img> tag :

inline level element

Embeds an image in html

```
<figure>
<img src="love.png" alt="Heart"
width="250" height="250"
loading="lazy"/>
<figcaption> A Red Heart </figcaption>
</figure>
```

empty element

images are not inserted into a web page, images are linked to web pages

default value : eager

 tag :

empty tag

```
<p>The start time for the
course is 9 pm.
<br>
However, the end time is
never fixed.
</p>
```

Block level element

Produces a line break in text

Don't use <br> to create margins between paragraphs, wrap them in
<p> tag and use CSS properties

Anchor tag :

defines a hyperlink

Attributes : target, download, href

```
<a href="https://www.google.com"
target="_blank">Search on Google
</a>
```

other values : _self , _blank , _parent , _top

An unvisited link is underlined and blue
A visited link is underlined and purple
An active link is underlined and red

List

allows to group a set of related items in lists

- Unordered list
- Ordered list
- Description list

Unordered list :

```
<ul>
<li>Java</li>
<li>C++</li>
<li>Dart</li>
</ul>
```

Block level element

Default symbol is disc shape bullet points

Output :

- Java
- C++
- Dart

type : circle, disc, square, triangle

Ordered list :

```
<ol>
<li>Java</li>
<li>C++</li>
<li>Dart</li>
</ol>
```

Block level element

Default symbol is numbers

Output :

1. Java
2. C++
3. Dart

type :
a for lowercase letters, A for uppercase letters, i for lowercase roman numerals, I for uppercase roman numerals, 1 for numbers (default)

Description list :

Block level element

```
<dl>
<dt> Dot Batch </dt>
<dd> Price is 5000 INR <dd>
<dt> Supreme Batch </dt>
<dd> Price is 3000 INR <dd>
<dl>
```

description list is a list of terms, with a description of each term

Output :

Dot Batch
    Price is 5000 INR
Supreme Batch
    Price is 3000 INR

dl : description list
dt : description term
dd : description detail

Understanding the boiler plate code of html5

Describes the type of doc : html version to our browser

Describes meta information about the site

```
<!DOCTYPE html>
<html lang="en">
<head>
<title>Hello, world!</title>
<meta charset="UTF-8" />
<meta name="viewport"
content="width=device-width,initial-
scale=1" />
<meta name="description" content="" />
</head>
<body>
<h1>Hello, world!</h1>

</body>
</html>
```

Contains all the data rendered by the browser

html tag : the root element

Note : Validate html code on https://validator.w3.org

The <head> is responsible for :

1. the document's title
   <title>About Me</title>
2. associated CSS files, fevicons, etc
   <link rel="stylesheet" type="text/css" href="style.css"
3. associated JavaScript files
   <script src="animations.js"></script>
4. the CHARSET being used (text encoding)
   <meta chat set="UTF-8">
5. Keywords, Descriptions (useful for SEO)
   <meta name="description" content="This pages describes about the founder of this company REDO and the investor holdings"

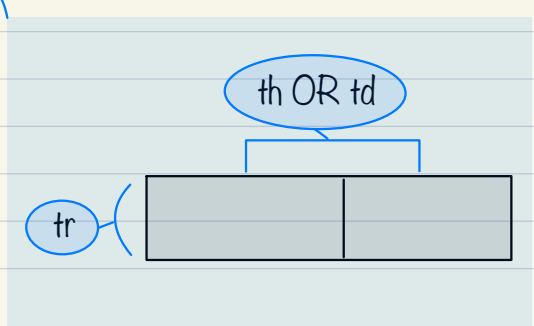
Tables in html

A table is a structured set of data made up of rows and columns.

```
<html>
    <head>
        <title>Table<title>
    </head>
    <body>
        <table>
            <caption> Student Data </caption>
            <thead>
                <tr>
                    <th> First Name </th>
                    <th> Last Name </th>
                </tr>
            </thead>
```

specifies a caption of the table

Wraps table head



```
<tbody>
    <tr>
        <td> ABC </td>
        <td> PQR </td>
    </tr>
    <tr>
        <td> XYZ </td>
        <td> TUV </td>
    </tr>
    <tr>
        <td colspan="2"> CSE DEPARTMENT ORIENTATION </td>
    </tr>
    <tr>
        <td> Day 1 </td>
        <td rowspan="2"> HEC BUILDING </td>
    </tr>
    <tr> <td> Day 2 </tr> </td>
</tbody>
<tfoot>
    <tr><td colspan="2">Emergency Contact : Andrew Sam </tr></td>
</tfoot>
</table>
</body>
</html>
```

Wraps table body

defines the no. of columns a table cell should span

defines the no. of rows a table cell should span

footer of table if any

Note : for table border, it is advised to achieve it using CSS

Therefore, for style use this piece of code inside < head> tag :

```
<style>
    table,
    td,
    th {
        border: 1px solid;
        border-collapse: collapse;
    }
</style>
```

Output :

Student Data

First Name	Second Name
ABC	PQR
XYZ	TUV
CSE DEPARTMENT ORIENTATION	
Day 1	HEC BUILDING
Day 2	
Emergency Contact : Andrew Sam	

Form in html

An html form is used to collect user input. The user input is often sent to server for processing.

Form : input tag, label tag, action attribute, method attribute, name attribute :

```
<html>
<head>
<title> HTML FORMS </title>
</head>
<body>
<h1> HTML FORMS </h1>

<form action="<script file url>" method='POST'>

<label for="username"> Username: </label>
<input type="text" id="username" name="username">
<br><br>
```

The action attribute defines the action to be performed when the form is submitted.

If the action attribute is omitted, the action is set to the current page.

should be same

```
<label for="password">Password: </label>
<input type="password" id="password" name="password">
<br><br>

<label for="email">Email: </label>
<input type="email" id="email" name="email">
<br><br>

<input type="submit" value="Sign in">

</form>
</body>
</html>
```

should be same

should be same

If the name attribute is omitted, the value of the input field will not be sent at all.

Output :

HTML FORMS

Username:

Password:

Email:

Sign in

placeholder attribute, required attribute :

- placeholder attribute specifies a short hint or some text in the input field

Username : your username here

- required attribute is a boolean attribute when present, it specifies that an input field must be filled out before submitting

Password :

Please fill out this field.

Example :

```
<label for="username"> Username </label>
<input type="text" id="username" placeholder="your username here" required>
<br><br>
```

Radio Button in Forms :

label ensures that even if we don't click specifically on radio button but on text, then also it will get selected.

The **name** attribute here makes sure that this radio button when clicked acts as follows : when one option is clicked other can't be selected ; that is only one out of three, not more than one

```
<p> Select your gender </p>

<label for="male"> Male </label>
<input type="radio" name="gender" id="male" value="Male"><br>

<label for="female"> Female </label>
<input type="radio" name="gender" id="female" value="Female"><br>

<label for="other"> Other </label>
<input type="radio" name="gender" id="other" value="Other"><br>
```

Output :

Select your gender

Male ◯

Female ◯

Other ◯

Checkbox :

id needs to be different in case of checkbox

```
<form action="<script file url>">

<input type="checkbox" value="maths" name="subject1" id="101">
<label for="101"> Maths </label>< br>

<input type="checkbox" value="physics" name="subject2" id="102">
<label for="102"> Physics </label>< br>

<input type="submit" value="Submit">
</form>
```

Output :

☐ Maths
☐ Physics

Submit

Dropdown :

```
<article><form>
<p>
<label for="coffee"> Favourite Coffee : </label>
<input type="text" name="coffee" id="coffee" list="coffee-list>

<datalist id="coffee-list">
<option value="coffee">
<option value="espresso">
<option value="americano">
<option value="other">
</datalist>
```

```
<p>
</form></article>
```

OR

```
<article><form>
<p>

<label for="coffee"> Favourite Coffee :</label>
<select name="coffee" id="coffee">
<option value="coffee"> Coffee </option>
<option value="espresso"> Espresso </option>
<option value="americano"> Americano </option>
<option value="other"> Coffee </option>
</select>

</p>
</form></article>
```

Output :

Favourite Coffee : ▽

With the first code : in this box we can even type to narrow our search in the dropdown

OR

Favourite Coffee : Coffee ▽

With the 2nd code : we can see we have a default value in the dropdown

---

Message Box and Submit :

```
<article><form>
<fieldset>

<legend> Send me a Note </legend>

<label for="message"> Your Message </label>
<br>
```

```
<textarea name="message" id="message" cols="30" rows="10"
    placeholder="type your message here ...">
</textarea>
</fieldset>
<br>

<input type="submit" value="Submit">

<button type="submit"> Submit </button>

<button type="reset"> Reset </button>

</form></article>
```

advised not to use this but use this one :

Output :

Send me a Note

Your Message

type your message ...

Submit Reset

formaction attribute and formmethod in <button> tag :

Used when we need two submit buttons in one form only

```
<form action="/action_page.php" method="GET">
  <label for="fname">First name:</label>
  <input type="text" id="fname" name="fname"><br><br>
  <label for="lname">Last name:</label>
  <input type="text" id="lname" name="lname"><br><br>
  <button type="submit">Submit</button>
  <button type="submit" formaction="/action_page2.php"
formmethod="POST">Submit to another page</button>
</form>
```

formmethod attribute :

The formmethod attribute specifies which HTTP method to use when sending the form-data. This attribute overrides the form's method attribute. The formmethod attribute is only used for buttons with type="submit". The form-data can be sent as URL variables (with method="get") or as HTTP post (with method="post").

formaction attribute :

The formaction attribute specifies where to send the form-data when a form is submitted. This attribute overrides the form's action attribute. The formaction attribute is only used for buttons with type="submit".

Note :

Notes on the "get" method:

- it appends the form-data to the URL in name/value pairs
- it is useful for form submissions where a user want to bookmark the result
- There is a limit to how much data you can place in a URL (varies between browsers), therefore, you cannot be sure that all of the form-data will be correctly transferred
- Never use the "get" method to pass sensitive information! (password or other sensitive information will be visible in the browser's address bar)

Notes on the "post" method:

- it sends the form-data as an HTTP post transaction
- Form submissions with the "post" method cannot be bookmarked
- it is more robust and secure than "get"
- it does not have size limitations

Example of FORM :

```
<html>
<head>
<title>HTML FORM </title>
</head>
<body>
<main>
<article id="contact">
<h2> Contact Me </h2>
<p> I'd really like to hear from you ! </p>
<form action="https://httpbin.org/get" method="GET" >
<fieldset>
<legend> Personal Info </legend>
<p>
<label for="firstname" > First Name : </label>
<input type="text" name="firstname" id="firstname" placeholder="Jane"
autocomplete="ON" required autofocus>
</p>
```

```
<p>
<label for="password" > Password : </label>
<input type="password" name="password" id="password" placeholder="your secret
password" required >
</p>
<p>
<label for="phone" > Phone : </label>
<input type="tel" name="phone" id="phone" pattern="[0-9]{3}-[0-9]{3}-[0-9]{4}"
required >
</p>
<p>
<label for="decade" > Favourite Decade: </label>
<input type="number" name="decade" id="decade" min="1950" max="2020"
step="10" value="1980" >
</p>
</fieldset>
</form>
</article>
</main>
</body>
</html>
```

Output :

Personal Info

Contact Me

I'd really like to hear from you !

First Name : Jane

Password : your secret password

Phone :

Favourite Decade : 1980

HTML Entities :

An html entity is a piece of text ("string") that begins with an ampersand (&) and ends with a semicolon (;). Entities are frequently used to display reserved characters (which would otherwise be interpreted as HTML code) and invisible characters (like non-breaking spaces). We can also use them in place of other characters that are difficult to type with a standard keyboard.

Result	Description	Entity Name	Entity Number
<	Less than	<	<
>	greater than	>	>
	non breaking space		
&	ampersand	&	&
"	double quotation mark	"	"
'	single quotation mark	'	'
$	Dollar sign	$	$
£	Pound sign	£	£
€	Euro sign	€	€
©	copyright	©	©
TM	trademark	&trade	™
®	registered trademark	®	®

Semantic Tags :

- ensures proper indexing by search engines
- Semantic Elements means elements with a meaning
- improving code readability for developers
- helps in appropriate screen rendering

Example :

1. <section>
2. <article>
3. <aside>
4. <details>
5. <figcaption>
6. <figure>
7. <footer>
8. <header>
9. <main>
10. <mark>
11. <nav>
12. <summary>
13. <time>

Adding Favicon in html :

A favicon is a small image displayed next to the page title in the browser tab.

```
<!DOCTYPE html>
<html>
<head>
  <title>My Page Title</title>
  <link rel="icon" type="image/x-icon" href="/images/favicon.ico">
</head>
<body>

<h1>This is a Heading</h1>
<p>This is a paragraph.</p>

</body>
</html>
```

Note : A favicon is a small image, so it should be a simple image with high contrast.

HTML Responsive Web Design :

Responsive web design is about creating web pages that look good on all devices! A responsive web design will automatically adjust for different screen sizes and viewports.

Responsive Web Design is about using HTML and CSS to automatically resize, hide, shrink, or enlarge, a website, to make it look good on all devices (desktops, tablets, and phones)

1. Setting The Viewport :

```
<meta name="viewport" content="width=device-width, initial-scale=1.0">
```

This will set the viewport of your page, which will give the browser instructions on how to control the page's dimensions and scaling.

2. Responsive Images :

Use responsive images that scale nicely to fit any browser size.

Using the width Property:
If the CSS width property is set to 100%, the image will be responsive and scale up and down

```
<img src="img_girl.jpg" style="width:100%;">
```

BUT

the image can be scaled up to be larger than its original size. A better solution, in many cases, will be to use the max-width property instead. If the max-width property is set to 100%, the image will scale down if it has to, but never scale up to be larger than its original size

```
<img src="img_girl.jpg" style="max-width:100%;height:auto;">
```

3. Show Different Images Depending on Browser Width :

The HTML <picture> element allows you to define different images for different browser window sizes.

```
<picture>
  <source srcset="img_smallflower.jpg" media="(max-width: 600px)">
  <source srcset="img_flowers.jpg" media="(max-width: 1500px)">
  <source srcset="flowers.jpg">
  <img src="img_smallflower.jpg" alt="Flowers">
</picture>
```

4. Responsive Text Size :

The text size can be set with a "vw" unit, which means the "viewport width". That way the text size will follow the size of the browser window

```
<h1 style="font-size:10vw">Hello World</h1>
```

5. Media Queries :

In addition to resize text and images, it is also common to use media queries in responsive web pages.
With media queries you can define completely different styles for different browser sizes.

```
<style>
.left, .right {
  float: left;
  width: 20%; /* The width is 20%, by default */
}
.main {
  float: left;
  width: 60%; /* The width is 60%, by default */
}
/* Use a media query to add a breakpoint at 800px: */
@media screen and (max-width: 800px) {
  .left, .main, .right {
    width: 100%; /* The width is 100%, when the viewport is 800px or smaller */
  }
}
</style>
```

HTML CHEATSHEET

1. <html lang="en"></html>
2. <!DOCTYPE html>
3. <head>

    <meta charset="UTF-8">

    <meta name="author" content="Satyam Kumar">

    <meta name="description" content="This is html cheatsheet">

    <title> My First html cheatsheet </title>

    <link rel="icon" href="something.png" type="image/x-icon">

    <style> ~~~~~~~~~~~</style>

    <link rel="stylesheet" href="<css file relative location>" type="text/css">

   </head>
4. <body></body>
5. <title></title>
6. <h1></h1> ... ... <h6></h6>
7.
8. <hr>
9. OR
10. </strong OR <bold></bold>
11. <abbr title="~~~~~~~~~~"> </abbr>

12. <address></address>
13.
14.
15.
16. <dl></dl> (description list)
17. <dt></dt> (description term)
18. <dd></dd> (description detail)
19.
20. <section></section>
21. <nav aria-label="primary-navigation"></nav>
22. <img src="~~~~~~" alt="~~~~~~" loading="lazy">
23. <figure></figure>
24. <figcaption></figcaption>
25. <main></main>
26. <header></header>
27. <footer></footer>
28. <aside></aside>
29. <mark></mark>
30. <time datetime="~~~("08:00" OR "PT3H")~~~></time>
31.
32.
33. <table></table>

34. <tr></tr>
35. <td></td>
36. <th></th>
37. <caption></caption>
38. <thead></thead>
39. <tfoot></tfoot>
40. <tbody></tbody>

41. <form action="~~~~~~" method="GET/POST">

```
    <label for="~~~~~~"></label>

    <input type="~~~~~~"
     name="~~~~~~"
     id="~~~~~~"
     placeholder="~~~~~~"
     autocomplete="ON/OFF"
     required autofocus pattern="~~~~~~"
     min="~~~~~~" max="~~~~~~" step="~~~~~~" value="~~~~~~">

</form>
```

SAME

search, tel, number, password, text, radio, checkbox, range, button, color, date, datetime-local, email, file, image, reset, submit, url, hidden

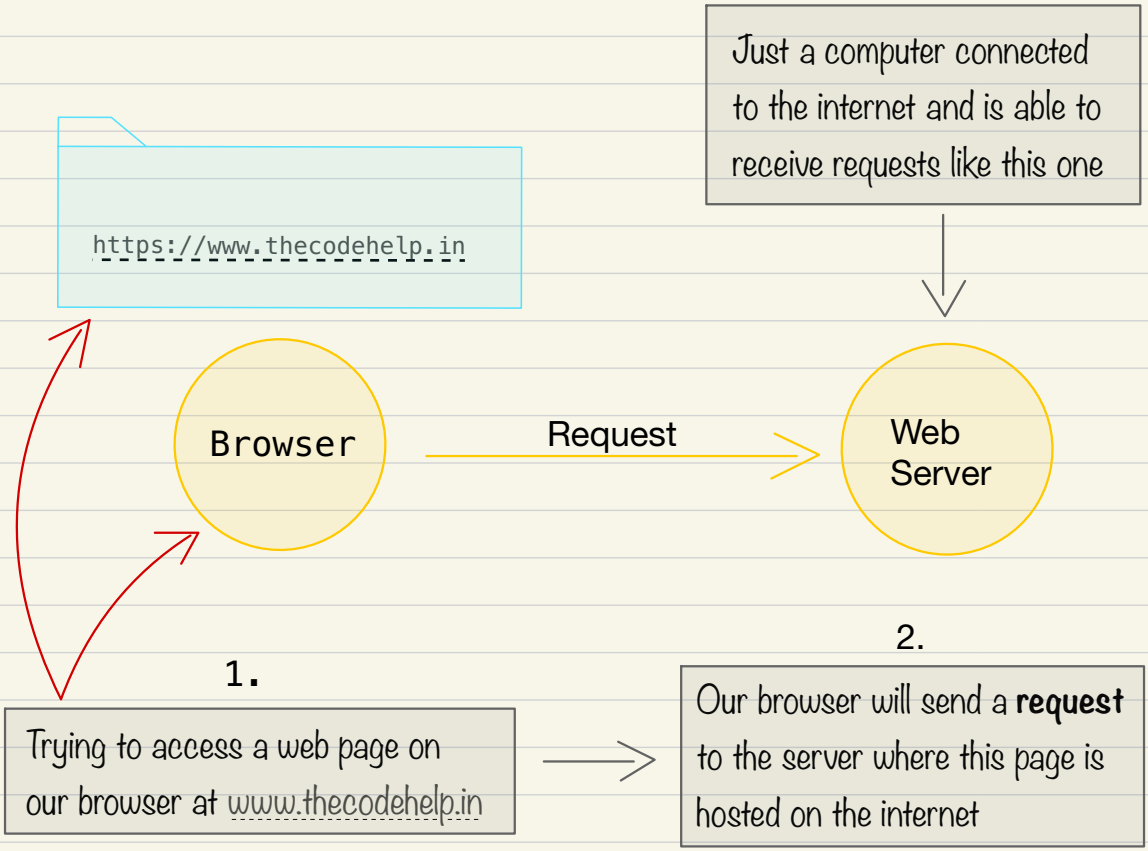
this is the value that will go to server

42. <fieldset></fieldset>
43. <legend></legend>
44.
 ---> line break
45. <details></details>
46. < summary></summary>
47. <mark></mark> ---> highlighter
48. <u></u> ---> underline tag
49. ---> superscript
50. ---> subscript
51. <del> ---> strike through
52. <small></small> ---> smaller than other text
53. <blockquote cite="<url>"> ~~~~~~~~~~~</blockquote> ---> quotation element
54. <cite> ~~~~~~~~~~~</cite> ---> citation element
55. <textarea name="~~~" id="~~~" cols="~~~" rows="~~~"></textarea>
56. <button type="submit" formaction="~~~" formmethod="~~~"></button>
57. <pre></pre> ---> any text within this tag will be displayed as it is with as much space as it is
58. <bdo dir="rtl"> This line will be written from right to left </bdo>
59. <map></map> AND <area> tag and usemap attribute on <img> tag
60. <picture></picture> AND <source> tag ---> always specify an <img> element as the last child element of <picture> element in case of any browser does not support <picture> or <source> tag

61. <code></code>
62. <kbd></kbd> ---> keyboard input
63. <samp></samp> ---> sample output from computer program
64. <var></var> ---> variable in programming

Static Website & Dynamic Website

Static Website : Whenever the files that make up a website are simply stored on the web server and are then sent to the browser as they are without any transformations, we then say we have a static website.



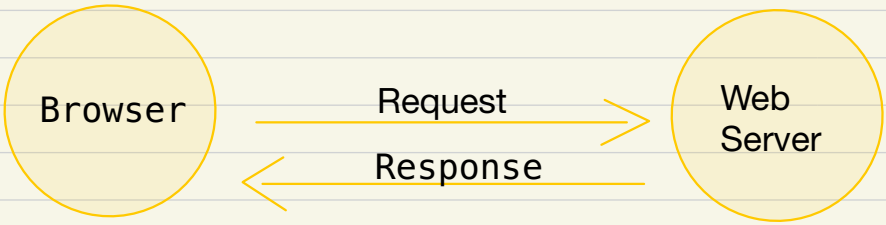
3.

When the server receives the request it will take up the files that make up the website

4.

server sends a response by sending all the files back to the browser required for that requested page

https://www.thecodehelp.in



But what if the website is a dynamic website ?

Dynamic Website :

a dynamic website is completely different from a static website because there is a lot of content changing all the time.

Thus, dynamic websites need a whole application running on the server side and also need a big database that contains all the contents that is being displayed on the website.

And to write applications that are actually executed on web servers, we use backend languages like Node.js . These languages take data out of database and assemble that data into final files that will then be sent to the browser as the response. In static website, the files are already done but in dynamic they first need to be generated.

And this whole process is back-end development.